ALFA BK UNIVERSITY
FACULTY OF INFORMATIONTECHNOLOGY
FACULTY O F MATHEMATICS AND COMPUTER SCIENCE
International Scientific Conference

Alfa BK Univerzitet

UDK: 811.163.41:004.655
DOI: 10.46793/ALFATECHproc25.171T
Original scientific paper

Syntactic and morphological data-base and algorithms for NLP of the Serbian language

Mirjana Tomic[1], Dejan Djukic[2]

Abstract: In this work, a method for automated generating grammatically correct sentences in Serbian language has been presented. This work presented a significant challenge, as the Serbian language is a highly inflected language, with complex word morphology, noun and adjective genders and declinations, verbal conjugations and concordance rules. The word components, such as word roots and their inflectional and morphological particles have been stored in JSON structures. The word components have been combined using custom produced software in Python programming language. The main software functions are the access to JSON data base of linguistic data, and the execution of algorithms for combining word parts into grammatically correct words, and words into syntactically and semantically correct sentences. The principal features of the software include morphological formation of verbs, nouns and adjectives, and combining these words with prepositions in a way to form sentences that appear to belong to the natural language use in Serbian language. The examples of generated sentences by this method show that such sentences, albeit somewhat simple, can be successfully generated by using this approach. The applications of the method presented here are numerous: from educational use, e.g. in language training, to more general research tools in the domain of natural language processing (NLP), not only for the Serbian language, but for a wider family of languages following complex grammatical rules, such as highly inflected, and morphologically complex languages.
Keywords: Natural language processing, NLP, automated natural language generation, NLG, Serbian language, data-base design, language and text synthesis, education, programming methodology

1. Introduction

Natural language generation (NLG), as has been defined for the domain of Artificial Intelligence (AI) as a part of Natural language processing (NLP), which has the aim of generating sensible phrases and sentences appearing as natural language. NLG is essentially an automated process for generating discourse. Whilst NLG can "write" it cannot "read". This task is being solved by the methods of Natural language understanding (NLU). Both NLG and NLU are sub-domains of NLP, which embrace both interpretation and generation of text, whether spoken or written:

- NLU is concerned with the semantical content of the natural language by taking into account the grammatical rules, the context, and the intent.
- NLP, taken in the narrowest sense, transforms a natural language discourse, a text, into predetermined language data structures.
- NLG generates natural-like discourse, text or speech, from the determined language data structures.

Thus NLG uses the recorded data, and through processes of filtering and transformation, it generates all kinds messages intended for humans for communicating with intelligent systems, but also all kinds of written texts, reports, and many more. At the same time, the theoretical advances in NLG make it a valuable tool not only in IT and computer domain, but also in psycho-linguistics and cognitive sciences [5].

At the present state of development of NLP methods and tools, it is important to extend the benefits of them also to highly inflected languages with complex word morphology, such as the Serbian language. Serbian language uses a rich set of morphological transformation rules, which comprise rules for gender and case declinations of nouns, pronouns and adjectives, rules for verb conjugations covering aspects of gender, person, tense and mode. All this causes for the generation of grammatically correct sentences in Serbian language to be a complex and highly involved task.

In this work, the authors propose a method of automated generation of sentences in Serbian language by using stored language data. The stored data comprises three separate data sets, one with vocabulary data, and another one with morphological rules such as case, gender, and tense suffixes, and one with algorithmic rules that sequence correctly formed words such as nouns, adjectives, verbs, adverbs and prepositions into sentences. The overall product of the application of the method are sensible and grammatically correct sentences. The data storage for this work has been performed using JSON storage structures. The algorithmic rules have been implemented as software written in Python programming language.

4. A NLG system for the Serbian language

2.1. Data structuring of the linguistic information

The word roots and the flexion suffixes are the base units for word creation. The data base of word roots (koreni.json) contains the basic forms of nouns, adjectives, verbs, adverbs, and prepositions. Each word root has been sorted by its grammatical category and by its basic morphological properties. These properties cover the grammatical gender, the number, and the case for nouns and adjectives, whilst the stored properties of the verbs are the tense, the person, and the number.
The database of flexion particles (nastavci.json) stores the suffixes necessary for the correct formation of words, according to the rules of grammar. The information structure of this file has

been designed so as to allow its use for various grammatical aspects of the works (gender, case, etc.). These data are then being read by the algorithm in the process of word and sentence generation.

2.2. Algorithmic generation of the linguistic information

The algorithm for word and sentence formation has been implemented in Python programming environment. It has comprises three main parts, reflecting the corresponding processing stages :

- **Reading of the stored data:** With the help of the function ***json.load()*** , the structured linguistic data stored in files ***word_roots.json*** and ***word_suffixes.json*** are being entered for further processing .
- **Word formation:** With the data read in step 1, function ***generisi_rec()*** performs unification of the word roots with appropriate suffixes. .
- **Sentence formation:** The function ***generisi_recenicu()*** produces sequence of the words formed in step 2. At this stage of development, the words are being combined at random, according to some predefined sequences of nouns, adjectives, verbs, adverbs and prepositions.
- **The formed sequences** of the words represent correct and complete sentences respecting the grammatical rules of the Serbian language.
- **Advantages of the approach and machine learning integration**: Adding machine learning modules could enable the system to autonomously acquire new syntactic and morphological patterns from provided examples. This integration would enhance both the system's flexibility and its

accuracy, making it capable of adapting to new language structures and improving the overall quality of sentence generation.

This approach combines rule-based processing with the potential for future machine learning enhancements, ensuring transparency in language generation while offering the possibility for adaptive improvements.

2.3. Python code samples

2.3.1. The JSON database and their purpose

The JSON database is structured to contain key information about different types of words (nouns, verbs, adjectives, etc.) and their grammatical properties (example in Table 1 and Table 2.).

Description	No.	Code / JSON Example
Example JSON database: **adjectives.json**	1.	```json
	2.	{
	3.	"crven": {
	4.	"rod": {
	5.	"muški": {
	6.	"jednina": {"nominativ": "crven", "akuzativ": "crvenog"},
	7.	"množina": {"nominativ": "crveni", "akuzativ": "crvene"}
	8.	},
	9.	"ženski": {
	10.	"jednina": {"nominativ": "crvena", "akuzativ": "crvenu"},
	11.	"množina": {"nominativ": "crvene", "akuzativ": "crvene"}
	12.	},
	13.	"srednji": {
	14.	"jednina": {"nominativ": "crveno", "akuzativ": "crveno"},
	15.	"množina": {"nominativ": "crvena", "akuzativ": "crvena"}
	16.	}
	17.	}
	18.	}
	19.	}

Table 1. JSON examples for adjective database

Description	No.	Code / JSON Example
Example JSON database: **nouns.json**	1.	```json
	2.	{
	3.	"ruža": {
	4.	"rod": "ženski",
	5.	"broj": ["jednina", "množina"],
	6.	"padeži": {
	7.	"nominativ": {"jednina": "ruža", "množina": "ruže"},
	8.	"genitiv": {"jednina": "ruže", "množina": "ruža"},
	9.	"dativ": {"jednina": "ruži", "množina": "ružama"},
	10.	"akuzativ": {"jednina": "ružu", "množina": "ruže"},
	11.	"instrumental": {"jednina": "ružom", "množina": "ružama"}
	12.	}
	13.	},
	14.	"zec": {
	15.	"rod": "muški",
	16.	"broj": ["jednina", "množina"],
	17.	"padeži": {
	18.	"nominativ": {"jednina": "zec", "množina": "zečevi"},
	19.	"genitiv": {"jednina": "zeca", "množina": "zečeva"},
	20.	"dativ": {"jednina": "zecu", "množina": "zečevima"},
	21.	"akuzativ": {"jednina": "zeca", "množina": "zečeve"},
	22.	"instrumental": {"jednina": "zecom", "množina": "zečevima"}
	23.	}
	24.	}
	25.	}

Table 2. JSON examples for noun database

2.3.2. JSON functions and their purpose

JSON (JavaScript Object Notation) is used for a structured representation of data that is easy to read and manipulate. It combines the base of the word with the corresponding grammatical features.

The function **generisi_rec()** generates the grammatically correct form of a word (eg noun, adjective or verb) using data about the grammatical properties of the word. It combines the base of the word (root) with the appropriate endings, depending on the gender, number and case of the word. (Table 3.)

Description	No.	Code / Example
Function: **genrisi_rec()**	1.	```python
	2.	def generisi_rec(base, gram_info):
	3.	broj = gram_info["broj"]
	4.	rod = gram_info["rod"]
	5.	padež = gram_info["padež"]
	6.	if base in gram_info["padeži"][padež][broj]:
	7.	return gram_info["padeži"][padež][broj]
	8.	return None
	9.	```
Example call: **generate_rec()**	10.	```python
	11.	# Example for a noun "ruža" (ženski rod, jednina, nominativ)
	12.	nouns = load_data("data/nouns.json")
	13.	print(generisi_rec("ruža", nouns["ruža"])) # Izlaz: "ruža"
	14.	```

Table 3. Example of a function for generating words

Description	No.	Code / Example
Function: **genrisi_recenicu()**	1.	```python
	2.	def generisi_recenicu():
	3.	nouns = load_data("data/nouns.json")
	4.	verbs = load_data("data/verbs.json")
	5.	adjectives = load_data("data/adjectives.json")
	6.	prepositions = load_data("data/prepositions.json")["predlozi"]
	7.	# Generating a subject
	8.	subject, subject_data = random.choice(list(nouns.items()))
	9.	subject_rod = subject_data["rod"]
	10.	subject_broj = random.choice(subject_data["broj"])
		subject_padež = "nominativ"
	11.	subject_adj = random.choice(list(adjectives.keys()))
	12.	subject_form = f"{adjectives[subject_adj]['rod'][subject_rod][subject_broj][subject_padež]} {subject}"
	13.	```
Example call: **genrisi _recenicu()**	14.	```python
	15.	# Loading data from JSON databases
	16.	print(generisi_recenicu())```

Table 4. Example of a function for generating sentences

The function **generisi_recenicu()** generates a complete sentence that is grammatically correct by combining subject, predicate and adverb clauses using data from JSON databases (nouns.json, verbs.json, adjectives.json, etc.). The result is a sentence that contains all the necessary parts according to the rules of the Serbian language. (Table 4.)

5. The results

The software has successfully produced sentences that comply with the basic grammatical rules of the Serbian language. These sentences demonstrate the ability of the system to generate semantically meaningful representations of common actions or descriptions of objects. The following table provides examples of sentences generated by the system (Table 5.):

No.	Generated sentence
1.	Crvena ruža cveta u zelenoj bašti.
2.	Crveno cveće cveta pored velikog drveta.
3.	Plavi zec mirno spava u velikom vrtu.
4.	Sjajna zvezda sija na velikom nebu.
5.	Mali pas trči prema plavom zecu.

Table 5: Examples of Generated Sentences

The generated sentences exhibit grammatically correct usage of key elements, including verbs, nouns, adjectives, adverbs, and prepositions. This confirms the effectiveness of the implemented morphological and syntactic rules within the system. The integration of JSON databases and Python algorithms ensures accurate selection and combination of word forms according to the grammar of the Serbian language.

While the sentences demonstrate grammatical correctness, they are relatively simple in structure, focusing on straightforward subject-predicate-object (SVO) or subject-predicate-adverbial patterns. This simplicity is intentional, as the system is currently optimized for generating basic sentences to validate the rules and data structure.

3.1. Observations and Current Limitations

3.1.1. Grammatical Accuracy:

The system consistently applies grammatical rules for cases, gender, and verb conjugations, which is particularly important for highly inflected languages like Serbian. However, the scope of the rules is limited to the present tense and does not yet cover complex tenses, moods, or irregular forms beyond a basic set.

3.1.2. Semantic Constraints:

Although the sentences are grammatically correct, they lack semantic depth. For instance, phrases like "Plavi zec mirno spava u velikom vrtu" may not always reflect realistic or contextually meaningful situations. This is due to the absence of semantic validation or contextual understanding in the current implementation.

3.1.3. Diversity of Sentence Structure:

The system predominantly generates simple declarative sentences. It does not yet support more complex structures, such as compound or interrogative sentences, which limits its application in scenarios requiring a richer variety of expressions.

3.2. Future Considerations for Improvement

To address these limitations, future iterations of the system should include:

- **Semantic rules**: Incorporating semantic constraints to generate contextually appropriate sentences.
- **Complex sentence structures**: Extending the algorithm to produce compound sentences, interrogative forms, and sentences with subordinated clauses.
- **Additional tenses and moods**: Expanding verb conjugation rules to include past and future tenses, as well as subjunctive and imperative moods.

Overall, the results demonstrate the potential of the NLG system for educational and research applications, providing a solid foundation for further development.

4. Applications

The results of the NLG system demonstrate its capability for various applications, particularly in education and research:

4.1. Educational use:

The system can serve as a valuable tool for practicing word morphology and sentence structures in Serbian. It is especially useful for native speakers or language learners in generating exercises for learning grammatical rules. By generating a large number of unique sentences, it reduces the teaching workload and enhances the efficiency of language learning.

4.2. Historical and linguistic studies:

With the inclusion of historical texts, the system can aid in studying how the Serbian language has evolved over time. This application could support linguistic research and the development of modern Serbian language resources.

4.3. Expansion of sentence complexity:

To increase its applicability, the system could be extended to generate more complex sentence structures, such as: compound sentences.sentences with subordinated clauses.interrogative and exclamatory sentences.

4.4. Contextual sentence generation:

Adding semantic controls would allow the system to generate sentences tailored to specific contexts, improving the relevance and usability of the output. Such context-aware generation could support tasks like creating conversational agents or language translation systems.

4.5. Machine learning integration:

By incorporating machine learning modules, the system could autonomously learn new syntactic and morphological patterns from example texts. This would enable more accurate and adaptive sentence generation, providing the basis for developing robust NLP applications for Serbian.

4.6. Flexibility for morphologically rich languages:

The system’s design is adaptable for other highly inflected and morphologically complex languages, broadening its potential application beyond Serbian.

4.7. General communication systems:

The system can be integrated into intelligent communication tools, such as chatbots or automated report generation systems, where grammatically correct sentence construction is essential.
This application spectrum highlights the system's potential not only for Serbian language processing but also as a general framework for other morphologically rich languages. With continued development, the system can bridge gaps in education, linguistics, and intelligent systems.

5. Conclusion

A system for the automated generation of grammatically correct sentences in the Serbian language has been developed, utilizing JSON for data storage and Python for algorithmic processing. The system effectively combines word roots with appropriate suffixes and sequences them into sentences following the grammatical rules of the Serbian language. The generated sentences, while simple, demonstrate the system's ability to handle the complex morphology and syntax of the Serbian language.
The primary goal of this system is its application in educational contexts, where it can aid in learning the Serbian language, its structures, and morphological rules. By generating a large number of unique sentences, it reduces the teaching workload and enhances learning efficiency.
However, the system's current limitations lie in the simplicity of the generated sentences. Future work could focus on implementing more complex sentence structures, such as compound sentences and subordinated clauses, as well as adding semantic controls for context-aware sentence generation. Incorporating machine learning models could further enhance the system's adaptability and accuracy, allowing it to autonomously learn new patterns.

This work provides a foundation for the development of natural language generation tools for morphologically complex languages like Serbian, with potential applications in education, linguistics, and intelligent communication systems.

6. References:

[1] D. Djukic and Z. Radovanovic, "Machine learning and theory of information in natural language processing," in *AIIT 2024 Proceedings*, Nov. 8, 2024, COBISS.SR-ID 158823945.
[2] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Pearson Prentice Hall, 2008.
[3] D. Mitrović, *Osnovi lingvističke gramatike srpskog jezika*. Beograd: Filološki fakultet, 2010.
[4] P. Piper and I. Klajn, *Normativna gramatika srpskog jezika*. Novi Sad: Matica srpska, 2013.
[5] S. Bird, E. Loper, and E. Klein, *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.
[6] Y. Goldberg, *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publishers, 2017.
[7] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 423–430.
[8] Ž. Bošković, "Clitics as Nonbranching Elements and the Linear Correspondence Axiom," *Linguistic Inquiry*, vol. 35, no. 2, pp. 329–340, 2004.
[9] J. Reisinger and M. Pasca, "Latent Variable Models of Concept-Attribute Attachment," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 620–628.
[10] N. Mikelić Preradović and J. Koehler, "Automatic Generation of Questions for Vocabulary Assessment," *Language Resources and Evaluation*, vol. 42, no. 2, pp. 161–173, 2008.
[11] L. Zlatić, "Morphosyntactic Features and the Structure of the Serbian Noun Phrase," in *Proceedings of the 22nd Annual Penn Linguistics Colloquium*, vol. 3, no. 1, pp. 145–159, 1997.
[12] B. Andrić, *Automatsko generisanje rečenica u srpskom jeziku: primena i izazovi*. Novi Sad: Univerzitet u Novom Sadu, 2018.
[13] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
[14] JSON, "JavaScript Object Notation (JSON) Data Interchange Format," 2023. [Online]. Available: https://www.json.org. [Accessed: 31-Jan-2025].
[15] Python Software Foundation, "Python Documentation," 2023. [Online]. Available: https://docs.python.org. [Accessed: 31-Jan-2025].

[16] J. Hajič, *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Prague: Charles University Press, 2004.
[17] J. Stojanović and S. Filipović, "Processing Morphologically Complex Words in Serbian," *Linguistica*, vol. 52, no. 1, pp. 25–44, 2012.
[18] O. M. Tomić, *Balkan Sprachbund Morpho-Syntactic Features*. Springer, 2006.
[19] D. Andor, C. Alberti, D. Weiss, *et al.*, "Globally Normalized Transition-Based Neural Networks," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 2442–2452, 2016.
[20] B. Plank and Ž. Agić, "Distant Supervision from Disparate Sources for Low-Resource Part-of-Speech Tagging," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 614–620, 2018.
[21] I. A. Sag, T. Wasow, and E. M. Bender, *Syntactic Theory: A Formal Introduction*. CSLI Publications, 2003.
[22] A. Kostić, *Word Frequency and Lexical Processing in Serbian*. Belgrade: Institute for Experimental Phonetics and Speech Pathology, 1991.

Contact information:

Mirjana Tomic[1]
Alfa BK University
Faculty of Information Technologies
Belgrade, Serbia
tomicmirjana3@gmail.com;
https://orcid.org/0009-0008-1152-1596

Dejan Đukić[2]
Alfa BK University
Faculty of Information Technologies
Belgrade, Serbia
dejan.djukic@alfa.edu.rs
https://orcid.org/0000-0001-7581-148X